



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Adress: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/676,374	09/30/2003	Markus Cherdron	09700.0064-00	3225
22852	7590	03/18/2008	EXAMINER	
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP 901 NEW YORK AVENUE, NW WASHINGTON, DC 20001-4413			INGBERG, TODD D	
ART UNIT	PAPER NUMBER			
	2193			
MAIL DATE	DELIVERY MODE			
03/18/2008	PAPER			

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/676,374	Applicant(s) CHERDRON ET AL.
	Examiner Todd Ingberg	Art Unit 2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
 - If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
 - Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED. (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 17 December 2007.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-6,8 and 10-21 is/are pending in the application.
- 4a) Of the above claim(s) 7 and 9 is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-6,8 and 10-21 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on 11/29/04 is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) Notice of References Cited (PTO-892)
 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
 3) Information Disclosure Statement(s) (PTO/SB/08)
 Paper No(s)/Mail Date _____
- 4) Interview Summary (PTO-413)
 Paper No(s)/Mail Date _____
- 5) Notice of Informal Patent Application
 6) Other: _____

DETAILED ACTION

Claims 1 – 6, 8 and 10 - 21 have been examined.

Claims 7 and 9 have been canceled.

Claims 1-6, 8 - 12, and 19 have been amended.

Specification

1. The amendment to the Specification has been entered..

Claim Objections

2. Objections to claims 12 and 19 has been overcome.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 1 – 6, 8 and 10 - 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over the Template Software product line in view of USPN# 5,872,973 **Mitchell et al**, issued February 16, 1999, filed October 26, 1995.

The **Template** Software product line contains:

The SNAP programming language

The Workflow Template

The Web Component

These three layered products work together.

The documentation sets for the products contains the following manuals.

SNAP released June 1997

SNAP Language Reference (Referred to as **REF** - Not used in this Office Action)

Using the SNAP Language (Referred to as **LANG**- Appendix A cited)

Using the SNAP Communication Component (Referred to as **COM**)

Using the SNAP Graphic User Interface Component (Referred to as **GUI** Not used in this Office Action)

Getting Started with SNAP (Referred to as **START** – portion cited)

Using the SNAP Display Editors (Referred to as **DISP** - Not used in this Office Action)

SNAP Class Library Reference (Referred to as **CLASS** - Not used in this Office Action)

Using the SNAP External Application Software Component (**EXT** - Not used in this Office Action)

Using the SNAP Development Environment (Referred to as **SNAP**)

SNAP Module Library Reference (Referred to as **MODU** Not used in this Office Action)

Using the SNAP Permanent Storage Component (Referred to as **PERM**)

Workflow released September 1997

Developing a WFT Workflow System (Referred to as **WFT**)

Using the WFT Development Environment (Referred to as **ENV**)

WFT Library Reference (Referred to as **WFTLIB** - Not used in this Office Action)

Web Component

Using the Web Component (Referred to as **WEB**)

Training Guides

SNAP Application Developer's Training Course (Referred to as **TRAINS** – Modules 1 and 10 provided)

Workflow Template Training Course (Referred to as **TRAINW** – Section A)

Since, these products work together they constitute a single reference and can be used as the basis for a rejection based on anticipated by a product offering. Furthermore, with the 1997 press release announcing version 8.0 these considered prior art under *In re Epstein* 31 USPQ2d 1817 (decided August 17, 1994) with a 1997 release date despite the 1998 copyright date.

Claim 1

Template anticipates a method for developing an application (WFT, pages 2-2 to 2-3, Introduction), the method comprising: defining file borders for development objects in a data model (SNAP, pages Chapter 8, Using the Project Editor – enables file borders from a high level – files for storing, WFT, Chapters 2 – 4, Enable one to design a workflow system at the application level (i.e. Data Flow level) see page 4-3); , wherein the data model includes a component class (SNAP, Chapter 3, page 3-40, defining attributes in classes is a data model – and component classes are the classes) and a model class associated with the component class (SNAP, Chapter 3, page 3-14 – shows the modeling of classes in the object model), and a controller class, associated with the component class, that associates a user interface to a business application model (LANG, Appendix A, explains the building, parsing, compiling and linking Applications – WFT as cited above show how to design the business application model which generates SNAP code. Furthermore, the ability to design User Interfaces is in Chapter 6 of ENV - LANG, Appendix A teaches the linking of the generated files); storing the development objects of the application in a file-based repository based on the file borders (TRAINW, Section A, page 3); and employing an API derived from the data model to access the development objects (WFT, Chapter 6, shows the Role objects called Applications in the WFT system – the previous sections described the flows (API) between the Applications). employing the API further comprises using tools that use the API to enable a user to perform a development operation (Mitchell, Col 2, Dependencies, Line 24 to Col 3, Line 63):: and wherein the development operation comprises enabling a user to refactor a copied one of the development objects to modify a key attribute of the copied one of the development objects (Mitchell, Figure 4 and col 25, lines 6 – 30).

In addition, to the inherent reuse of object oriented technology as taught by Template. Mitchell teaches tools specifically for Refactoring. One of ordinary skill in the art at the time of invention would be motivated to combine the teachings of Template and Mitchell, because specific tools to perform Refactoring enable software development organizations to be more cost effective.

Examiner's Note

No patentable weight given to the naming of a class (controller).

Claim 2

The method of claim 1, further comprising caching the development objects in a local cache repository (SNAP, pages 3-10, the classes represent objects and are reusable, NOTE: local cache is not defined to be traditional meaning of cache memory).

Claim 3

The method of claim 1, wherein defining the file borders comprises identifying one of the development objects as a main development object to be included in a file with any development objects that are defined in the data model to be children objects of the main development object that are not identified as main development objects (SNAP, Chapter 3, Object Model Editor, see pages 3-6 inheritance lines and page 3-9 and Mitchell as per claim 1).

Claim 4

The method of claim 3, further comprising storing in the file user-defined code associated with the main development object (SNAP, page 3-44 to 3-47, Functions (methods) and Mitchell as per claim 1).

Claim 5

The method of claim 3, further comprising storing in the file a reference to another development object stored in another file (ENV, pages 2-1 to 2-7 and SNAP, Chapter 8 and Mitchell as per claim 1).

Claim 6

The method of claim 1, further comprising enabling a user to define a source path for one of the development objects As per claim 1.

Claim 8

The method of claim 1 wherein the development operation includes a copy and paste operation (SNAP, page 3-32 and Mitchell as per claim 1).

Claim 10

The method of claim 1 further comprising enabling a user to define a scope of the refactor (TRAINW, Section A, page 3, Shows the Template concept in view of the rejection for claim 1 – objects and classes refactored – scope Vertical Templates – Template).

Claim 11

The method of claim 1 wherein the development operation includes storing translatable text separate from the development objects (SNAP, page 6-10, Class Definition files (CD) more specific references to CD files throughout the reference).

Claim 12

Template anticipates a method for developing applications, the method comprising: generating a data model for an application (as per claim 1), the data model being implemented in a language that includes a customizable extension (Template product line is object technology), the data model including a feature defined using the customizable extension (SNAP, Chapter 3, Object Model Editor, see pages 3-6 inheritance lines and page 3-9); , wherein the data model includes a component class (SNAP, Chapter 3, page 3-40, defining attributes in classes is a data model – and component classes are the classes) and a model class associated with the component class (SNAP, Chapter 3, page 3-14 – shows the modeling of classes in the object model), and a

controller class, associated with the component class, that associates a user interface to a business application model(LANG, Appendix A, explains the building, parsing, compiling and linking Applications – WFT as cited above show how to design the business application model which generates SNAP code. Furthermore, the ability to design User Interfaces is in Chapter 6 of ENV - LANG, Appendix A teaches the linking of the generated files); ; deriving an API from the data model, the API incorporating the feature; and enforcing constraints specified in the data model by employing the derived API during development of the application (WFT, pages 4-20 to 4-28, Rules). : and enabling a user to access development objects of the application using the API, wherein tools that use the API enable a user to perform a development operation (Mitchell, Col 2, Dependencies, Line 24 to Col 3, Line 63); wherein the development operation comprises enabling a user to refactor a copied one of the development objects to modify a key attribute of the copied one of the development objects (Mitchell, Figure 4 and col 25, lines 6 – 30). In addition, to the inherent reuse of object oriented technology as taught by Template. Mitchell teaches tools specifically for Refactoring. One of ordinary skill in the art at the time of invention would be motivated to combine the teachings of Template and Mitchell, because specific tools to perform Refactoring enable software development organizations to be more cost effective.

Claim 13

The method of claim 12, wherein the feature comprises an indication used to implement a file border. (Mitchell, col 12, lines 49 - 70).

Claim 14

The method of claim 12, wherein the feature comprises an indication used to implement a platform-specific feature. (SNAP, Chapter 8, page 8-26 – UNIX and pages 2-4 to 2-5).

Claim 15

The method of claim 12, wherein the feature comprises an indication representing translatable text. (SNAP, page 6-10, Class Definition files (CD) more specific references to CD files throughout the reference).

Claim 16

The method of claim 12, wherein the feature comprises an indication representing that an aggregation in the data model is ordered. (SNAP, Chapter 3, Object Model Editor, see pages 3-6 inheritance lines and page 3-9).

Claim 17

The method of claim 12, wherein the feature comprises an indication representing a singular name. (SNAP, page 3-11, New class - class name)

Claim 18

The method of claim 12, wherein the feature comprises an indication representing that an attribute in the data model is nullable. (SNAP, page 3-40, Attributes – Default).

Claim 19

Template anticipates a computer program product containing instructions which when executed on a processor, form a system for developing an application (WFT, pages 2-2 to 2-3, Introduction), the system comprising: a repository storing development objects using file borders defined in a data model (SNAP, pages Chapter 8, Using the Project Editor – enables file borders from a high level – files for storing, WFT, Chapters 2 – 4, Enable one to design a workflow system at the application level (i.e. Data Flow level) see page 4-3); , wherein the data model includes a component class (SNAP, Chapter 3, page 3-40, defining attributes in classes is a data model – and component classes are the classes) and a model class associated with the component class (SNAP, Chapter 3, page 3-14 – shows the modeling of classes in the object model) , and a controller class, associated with the component class, that associates a user interface to a business application model(LANG, Appendix A, explains the building, parsing, compiling and linking Applications – WFT as cited above show how to design the business application model which generates SNAP code. Furthermore, the ability to design User Interfaces is in Chapter 6 of ENV - LANG, Appendix A teaches the linking of the generated files);; a local development cache for caching the development objects from the repository (SNAP, pages 3-10, the classes represent objects and are reusable, NOTE: local cache is not defined to be traditional meaning of cache memory) ; an API derived from the data model (WFT, pages 2-4, shows the flows between the different Roles from the concept drawing – the implementation on page , WFT, Chapter 7 shows different interactions among the objects); and a user interface development tool that uses the API to access the development objects (WFT, Chapter 6, shows the Role objects called Applications in the WFT system – the previous sections described the flows (API) between the Applications). a second user interface development tool for enabling a user to refactor a copied one of the development objects to modify a key attribute of the copied one of the development objects (Mitchell, figure 4, col 25, lines 6-17). Template teaches an object oriented computer aided software engineering tool that inherently has reuse of code. But Template does not teach a user interface specifically for Refactoring. It is Mitchell who teaches a specific user interface for refactoring (Mitchell, Figure 4). Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to combine Template and Mitchell, because specific tools to enhance object oriented codes refactoring provide a means of introducing efficiency of programming and cut cost of development.

Claim 20

The system of claim 19, further comprising a repository server that includes the repository (WFT, pages 7-19).

Claim 21

The system of claim 19, wherein the user interface development tool comprises one of a project browser (as per claim 19), an application modeler, a view designer, a controller and context editor, and a model editor (ENV, Chapter 3, WDE).

Response to Arguments

5. Applicant's arguments with respect to claims 1 – 6, 8 and 10 - 21 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

6. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Todd Ingberg whose telephone number is (571) 272-3723. The examiner can normally be reached on during the work week..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on (571) 272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Todd Ingberg/
Primary Examiner

TI